**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота № 5**

з дисципліни «Системне програмування»

на тему

«Арифметичні і логічні операції з цілими числами. Масиви.»

Виконав: Перевірив:

студент групи ІМ-22 доцент

Куц Іван Васильович Павлов В.Г.

Номер у списку групи: 12

**Київ 2024**

**Мета роботи**

Вивчення арифметичних і логічних команд Асемблера і здобуття навичок виконання розрахунків з елементами масивів.

**Скріншоти виконання програми**

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

**Скріншоти виконання програми при 0 у знаменнику**

**Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис**

**Контрольні розрахунки**

Вираз =

1. a = 2, b = 3, c = 9, d = 8

Знаки ; = = 2

1. a = -2, b = 2, c = 96, d = 32

Знаки ; = = -2

1. a = -2, b = 2, c = 676, d = 168

Знаки ; = = -1

1. a = -2, b = 2, c = 668, d = 168

Знаки ; = = 1

1. a = -2, b = 2, c = 8, d = 2

;

**Лістинг програми**

C ; «««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««

C

00000000 .data

00000000 53 79 73 74 65 labTitle db "System programming, lab 5, Ivan Kuts, Variant 12", 0

6D 20 70 72

6F 67 72 61

6D 6D 69 6E

67 2C 20 6C

61 62 20 35

2C 20 49 76

61 6E 20 4B

75 74 73 2C

20 56 61 72

69 61 6E 74

20 31 32 00

00000031 46 6F 72 6D 75 withResultsMessageTemplate db "Formula: (41 - d / 4 - 1 )/( c / b + a \* d )", 10, 10,

6C 61 3A 20

28 34 31 20

2D 20 64 20

2F 20 34 20

2D 20 31 20

29 2F 28 20

63 20 2F 20

62 20 2B 20

61 20 2A 20

64 20 29 0A

0A 49 6E 69

74 69 61 6C

20 72 65 73

75 6C 74 20

3D 20 28 34

31 20 2D 20

25 69 20 2F

20 34 20 2D

20 31 20 29

2F 28 20 25

69 20 2F 20

25 69 20 2B

20 25 69 20

2A 20 25 69

20 29 20 3D

20 25 69 0A

4D 6F 64 69

66 69 65 64

20 72 65 73

75 6C 74 20

3D 20 25 69

0A 77 68 65

72 65 3A 20

61 20 3D 20

25 69 2C 20

62 20 3D 20

25 69 2C 20

63 20 3D 20

25 69 2C 20

64 20 3D 20

25 69 0A 64

65 6E 6F 6D

69 6E 61 74

6F 72 20 3D

20 25 69 2C

20 6E 75 6D

65 72 61 74

6F 72 20 3D

20 25 69 0A

0A 00

"Initial result = (41 - %i / 4 - 1 )/( %i / %i + %i \* %i ) = %i", 10,

"Modified result = %i", 10,

"where: a = %i, b = %i, c = %i, d = %i", 10,

"denominator = %i, numerator = %i", 10, 10, 0

000000FC 46 6F 72 6D 75 errorMessageTemplate db "Formula: (41 - d / 4 - 1 )/( c / b + a \* d )", 10, 10,

6C 61 3A 20

28 34 31 20

2D 20 64 20

2F 20 34 20

2D 20 31 20

29 2F 28 20

63 20 2F 20

62 20 2B 20

61 20 2A 20

64 20 29 0A

0A 49 43 4F

52 52 45 43

54 20 44 45

4E 4F 4D 49

4E 41 54 4F

52 20 50 41

52 41 4D 45

54 45 52 53

0A 28 20 25

69 20 2F 20

25 69 20 2B

20 25 69 20

2A 20 25 69

20 29 20 3D

3D 20 30 0A

44 49 56 49

53 49 4F 4E

20 42 59 20

30 20 49 53

20 4E 4F 54

20 41 4C 4C

4F 57 45 44

0A 47 4F 20

41 4E 44 20

4C 45 52 41

4E 20 4D 41

54 48 0A 0A

00

"ICORRECT DENOMINATOR PARAMETERS", 10,

"( %i / %i + %i \* %i ) == 0", 10,

"DIVISION BY 0 IS NOT ALLOWED", 10,

"GO AND LERAN MATH", 10, 10, 0

00000196 00000002 arrayA dd 2, -2, -2, -2, -2

FFFFFFFE

FFFFFFFE

FFFFFFFE

FFFFFFFE

000001AA 00000003 arrayB dd 3, 2, 2, 2, 2

00000002

00000002

00000002

00000002

000001BE 00000009 arrayC dd 9, 96, 676, 668, 8

00000060

000002A4

0000029C

00000008

000001D2 00000008 arrayD dd 8, 32, 168, 168, 2

00000020

000000A8

000000A8

00000002

00000000 .data?

00000000 00000005 [ finalValue dd 5 dup(?)

00000000

]

00000014 00000001 [ numerator dd 1 dup(?)

00000000

]

00000018 00000001 [ denominator dd 1 dup(?)

00000000

]

0000001C 00000200 [ labMessage db 512 dup(?)

00

]

00000000 .code

00000000 lab5:

00000000 BE 00000000 mov esi, 0

.while esi < 5

;; denominator

0000000A 8B 04 B5 mov eax, dword ptr [arrayC + esi\*4] ;; Load c into eax

000001BE R

00000011 8B 1C B5 mov ebx, dword ptr [arrayB + esi\*4] ;; Load b into ebx

000001AA R

00000018 33 D2 xor edx, edx ;; Clear edx for division

0000001A F7 F3 div ebx ;; c / b => Result in eax, remainder in edx

0000001C 8B C8 mov ecx, eax ;; Store c/b in ecx temporarily

0000001E 8B 04 B5 mov eax, dword ptr [arrayA + esi\*4] ;; Load a into eax

00000196 R

00000025 8B 14 B5 mov edx, dword ptr [arrayD + esi\*4] ;; Load d into edx

000001D2 R

0000002C 0F AF C2 imul eax, edx ;; a \* d => Result in eax

0000002F 03 C1 add eax, ecx ;; eax = (c/b + a\*d)

00000031 A3 00000018 R mov dword ptr [denominator], eax ;; Store the result in 'denominator'

.if denominator == 0

invoke wsprintf, addr labMessage, addr errorMessageTemplate,

arrayC[esi \* 4], arrayB[esi \* 4], arrayA[esi \* 4], arrayD[esi \* 4]

invoke MessageBox, 0, offset labMessage, offset labTitle, 0

.else

;; numerator calculation for 40 - d / 4

00000085 B9 00000004 mov ecx, 4

0000008A 8B 04 B5 mov eax, dword ptr [arrayD + esi \* 4] ;; Load d into eax

000001D2 R

00000091 99 cdq ;; Sign extend eax into edx:eax for division

00000092 F7 F9 idiv ecx ;; Divide edx:eax by ecx, result in eax

00000094 B9 00000028 mov ecx, 40

00000099 2B C8 sub ecx, eax ;; Subtract quotient from 40

0000009B 89 0D 00000014 R mov dword ptr [numerator], ecx ;; Store the result in numerator

000000A1 A1 00000014 R mov eax, numerator ;; Loading numerator into eax register

000000A6 8B 0D 00000018 R mov ecx, denominator ;; Loading denominator into ecx register

000000AC 99 cdq

000000AD F7 F9 idiv ecx ;;

000000AF A3 00000000 R mov finalValue, eax ;; Storing result in a variable

000000B4 A9 00000001 test eax, 1 ;; check if result is odd or even

000000B9 75 0C jnz oddResult

000000BB 74 00 jz evenResult

000000BD evenResult:

000000BD B9 00000002 mov ecx, 2

000000C2 99 cdq

000000C3 F7 F9 idiv ecx ;; finalValue / 2

000000C5 EB 05 jmp message

000000C7 oddResult:

000000C7 6B C0 05 imul eax, 5 ;; finalValue \* 5

000000CA EB 00 jmp message

000000CC message:

invoke wsprintf, addr labMessage, addr withResultsMessageTemplate,

arrayD[esi \* 4], arrayC[esi \* 4], arrayB[esi \* 4], arrayA[esi \* 4], arrayD[esi \* 4],

finalValue, eax,

arrayA[esi \* 4], arrayB[esi \* 4], arrayC[esi \* 4], arrayD[esi \* 4],

denominator, numerator

invoke MessageBox, 0, offset labMessage, offset labTitle, 0

.endif

00000143 C6 05 0000001C R mov labMessage, 0h

00

0000014A 46 inc esi

.endw

end lab5

**Висновок**

Виконуючи цю лабораторну роботу, я ознайомився з виконанням арифметичних операцій та використанням масивів у ASM32.

Роботу програми було перевірено на 5 наборах змінних. До змінних виставлялись такі вимоги:

змінні не можуть дорівнювати 0 та 1.

Оскільки такий вид роботи я виконую вперше, після виконання я вручну провів обчислення, які попередньо були запрограмовані. Після порівняння отриманих значень можна зробити висновок, що обрахунки були проведені коректно. Отримані результати збігаються з очікуваними

Також у лабораторній було оброблено випадок з діленням на 0 у знаменнику. У такому випадку програма показує повідомлення про неможливість обрахунку виразу.